Modeling crisis management process from goals to scenarios

Elena Kushnareva, Irina Rychkova, Rébecca Deneckére, Bénédicte Le Grand

Modeling crisis management process from goals to scenarios

Elena Kushnareva, Irina Rychkova, Rébecca Deneckére, and Bénédicte Le Grand

University Paris 1 Panthéon-Sorbonne,
12, Place de Panthéon, 75005 Paris, France
elena.kushnareva@malix.univ-paris1.fr,{irina.rychkova,benedicte.le-grand,rebecca.deneckere}@univ-paris1.fr
http://www.univ-paris1.fr/

Abstract. Process manager plays the central role in crisis management. In order to capture the intentions of the process manager, the process should be specified at a strategic level. In order to analyze how these intentions are fulfilled during the process execution, the process should be specified at an operational level. Whereas the variety of techniques for goal modeling and process modeling is presented in the market, possibility to design a process seamlessly, from intentions (process goals) to executable scenarios, remains a challenging task. In this paper, we introduce an approach for modeling and simulating a crisis management process from goals to scenarios. We consider an example of flood management process specified for floods on Oka River in the Moscow region in Russia. In order to specify the intentions behind the flood management process, we use MAP formalism. For representing the process at the operational level, we use Statecharts formalism. To align the strategic and operation process levels, we translate the MAP model of flood management process to statecharts. We simulate the flood management process, showing how the process goals defined on the strategic level can be achieved by various scenarios executed in the operational level.

Keywords: Statecharts, MAP, process flexibility, Intentional Modeling, State-based Modeling

1 Introduction

Crisis management process is an example of a knowledge-intensive process (KiPs) [5]: it strongly depends on the situation (context) and tacit knowledge of human actors plays the central role in this process. On the other hand, crisis management has to comply with federal regulations such as the Emergency Management Guidelines. These characteristics make specification and implementation of crisis management solutions challenging.

In this paper, we consider the case of flood management process specified for floods on Oka River in the Moscow region in Russia. This process is implemented as a part of COS Operation Center (COSOC) - a smart city solution developed by COS&HT [8]. The existing COSOC solution was designed following traditional workflow-based approach: the flood management process is specified as a sequence of activities that have to be executed according to the current conditions (i.e., water level and status of the flooded regions) and in compliance with the Emergency Management Guidelines defined by the Ministry of Emergency Situations (MES) in Russia.

The experience shows that execution of workflows proposed by COSOC can often be problematic due to unforeseen circumstances (e.g., lack of resources, disrupted traffic, etc.) Current system provides only limited support for process flexibility and in order to implement an alternative scenario, the process manager often switches to "off-line" mode.

To improve the process flexibility and adaptability:

1) Process specification should capture its intentional perspective, or should answer a question "WHY do we carry out an activity/procedure?". Understanding intentions behind the guidelines will help to define alternative scenarios in the situations when the default scenario cannot be implemented [16][1].

2) Process specification should focus on WHAT has to be done instead of HOW it must be done [6]. In this case, only the process outcomes need to be fixed at design while concrete procedures or activities leading to these outcomes can be chosen by the process manager at run-time.

In this paper, we introduce an approach for modeling crisis management process from goals to executable scenarios. We use MAP to reason about intentions behind the flood management process (strategic level). MAP is a goal-oriented modeling language introduced by Rolland in [11][12][14].

We use statecharts formalism for representing the flood management process at the operational level. Statecharts is a state-oriented modeling formalism defined by Harel [7].

To align the strategic and operation process levels we translate the MAP model of flood management process to the statecharts.

Statecharts specifications are executable. We simulate the flood management process, showing how the process goals defined on the strategic level can be achieved by various scenarios executed in the operational level.t

The remainder of the paper is organized as follows: in Section 2 we introduce the case study of a flood management process and specify this process on the operational level using statecharts; in Section 3 we specify a goal model for the flood management process; in Section 4 we translate MAP model to statecharts; in Section 5 we illustrate how the complete model of the flood management process - strategic and operational levels - can be simulated in YAKINDU Statecharts tool; in Section 6 we draw our conclusions.

2 Flood Management Process: Operational level

In this section we briefly introduce our case study - the flood management process; we show how this process can be modeled with Statecharts formalism.

2.1 Flood Management Process

Floods on the Oka River in the Moscow region are seasonal events caused by an increase in the flow of the river, provoked by intensive snow melting during the spring months. Cities built along the Oka River are confronted to the risk of flooding and can expect important damages, affecting thousands of people. Floods on Oka also represent substantial risks for the critical infrastructure facilities situated in the area: a railway bridge, a pontoon road bridge, an electric power plant, etc.

Along with other types of crisis, the flood management process has to comply with the Emergency Management Guidelines [15] defined by MES. This document prescribes the activities that have to be carried before, during and after crisis situations by different public services and agencies of the city.

As specified in Section 1, the flood management process is implemented as a part of COSOC - a process-aware information system used by a government to manage the variety of cross-domain operations within the city.

The functions of COSOC can be roughly divided into three groups: (i) data collection and visualization, (ii) analysis of the situation and decision making and (iii) triggering response processes.

The COSOC process manager is a member of the city administration who is responsible for monitoring the situation and handling emerging issues. He/she can accept or decline the solution proposed by the system; when a workflow is triggered, he/she monitors its execution and intervenes when decision-making is required.

When the problematic situation is resolved, the process manager can provide feedback to the system: request for process improvement, modification of monitored parameters list, etc.

In our previous work [10], we examined the BPMN specification of the flood management process designed for COSOC and showed that the capacity of PAIS to support flexibility of the process is inherent to the underlying process modeling paradigm.

In order to provide the flexibility of the process, we design a state-oriented model of the flood management process in the YAKINDU Statechart Tool (SCT) using the formalism of statecharts.

2.2 Statecharts model

The statecharts formalism specifies hierarchical state machines (HSM) and extends classical finite-state machines (FSM) by providing: *depth* (the possibility to model states at multiple hierarchical levels); *orthogonality* (the possibility to model concurrent or independent sub-machines within one state machine); *broadcast communication* (the possibility to synchronize multiple concurrent sub-machines via events).

The statecharts model (Fig. 1) describes the process with a set of states (e.g., *Alert*, *Emergency*, *Restoring Normal Functioning*, etc.) and transitions between them. Process execution starts at an initial state and terminates at a final state.

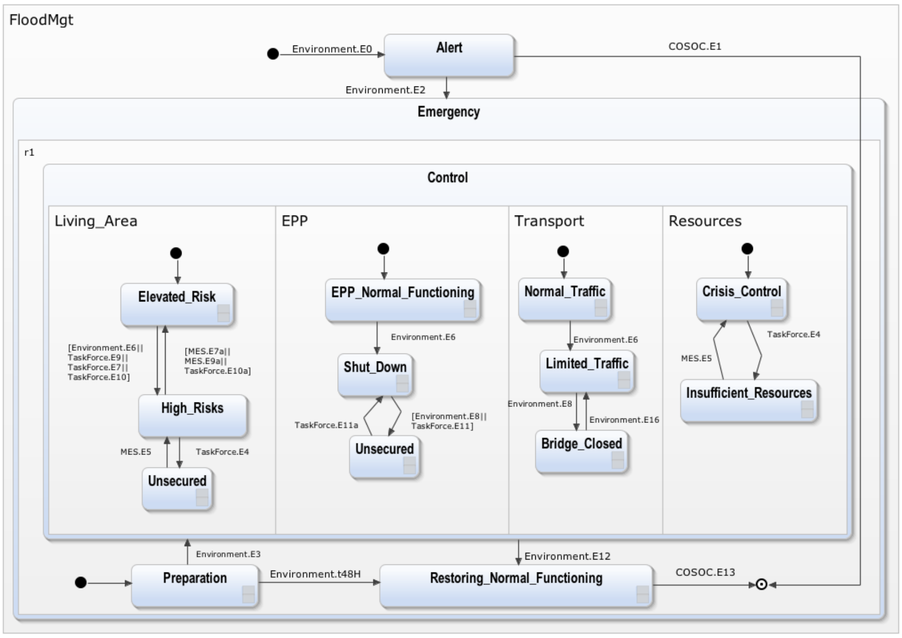

Fig. 1. Statechart model of a flood management process

The sequence of states and transitions that leads from the initial state to the final state can be seen as a process scenario.

In statecharts (unlike Petri Nets, where transitions are associated with the execution of an activity) each state transition can be triggered by a specific event or combination of events. For example, the event *E2: water level h>10cm and keeps rising* triggers a transition from *Alert* to *Emergency*. The activities producing these events can be selected at run-time. We call this *deferred binding*.

As a result, a process specification is divided into two parts: the state-transition part, defined with a set of states and transitions between states and their triggering events, and the activity part, defined by a list of activities specified by their preconditions and outcomes. The process enactment can be seen as a dynamic selection of activities to produce some outcomes (events) that make the process progress towards its (desired) final state.

3 Flood Management Process: Strategic level

In this section we explain our choice of modeling formalism, introduce the concept of MAP, illustrated on the flood management process example, and propose a procedure for transforming a MAP to a Statechart model.

3.1 Choosing the formalism

Processes may be formalized in an intentional way in goal-modeling approaches to model the processes according to the purpose of the actors/organizations. We quote among them i* [17][18], KAOS [3] and MAP. We choose the MAP modeling language in our approach because it allows formalizing flexible processes with high level organizational intentions. It supports variability for the goals and offers the possibility to follow different strategies by focusing on the intentional aspect when enacting methodological processes [4]. i* has an operational semantic for the tasks but not for the goals and it is not used to model strategic goals; it is not designed to be a variable framework, therefore it does not afford a high level of flexibility. As for KAOS, it supports variability and have a well-structured semantic but is less involved in the intentional aspect of IS actors. Furthermore, KAOS has a rigid task-decomposition; modeling complex intentional processes is then difficult [13].

3.2 MAP model

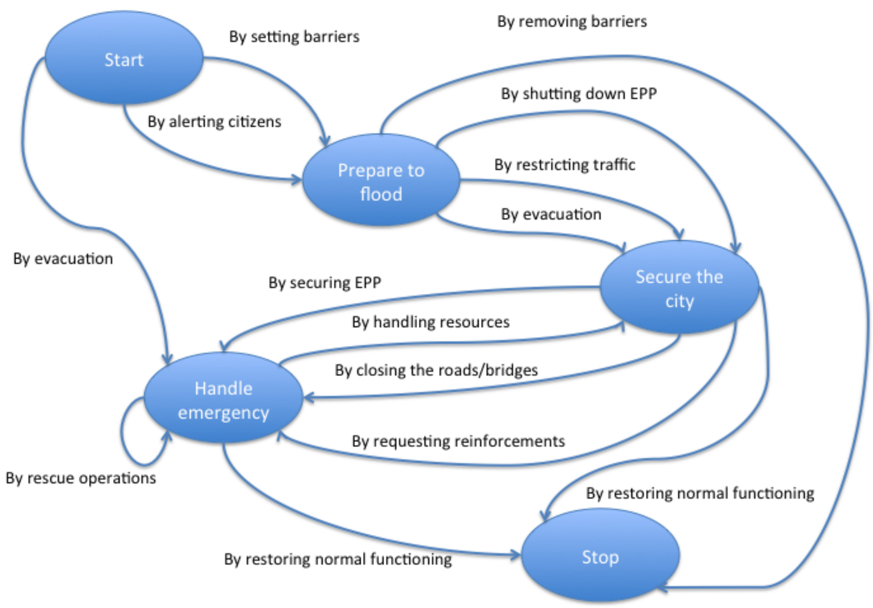

Fig. 2. MAP model of the flood management process

MAP specifies processes in a flexible way by focusing on the intentions and the different ways to fulfill them, according to the context of each actor. A MAP model is presented as a graph which nodes represent *intentions* and edges represent *strategies*. An edge is defined between two nodes where its related strategy can be used to achieve the intended target node. There may be several edges entering a node representing all the strategies to fulfill an intention. A *section* is a triplet <source intention, strategy, target intention>, which represents a particular process step to achieve an intention (the target intention) from a specific situation (the source intention) following a particular technique (the strategy). The MAP meta-model has been validated in several domains as IS engineering [14], requirement engineering [13], method engineering [9], enterprise knowledge development [2], etc.

The MAP corresponding to the flood management process is presented in Fig. 2.

There are three intentions specified on this level: *Prepare to flood*, *Secure the city* and *Handle emergency*. Each intention can be fulfilled by several strategies. For example, *Handle emergency* intention can be reached either from the initial intention (*Start*) - in case of need for emergency evacuation of the citizens before the preparations finished, or from the *Secure the city* intention - in case of lack of resources, or getting the 'Electric Power Plant (EPP) is flooded' alert. The final intention (Stop) corresponds to the process termination and can be attained either *By removing barriers*, or *By restoring normal functioning* of objects of the city infrastructure. The flood management MAP model also contains a recursive section <*Handle emergency, By rescue operations, Handle emergency*>, which represents the maintenance of a need in emergency handling while receiving the rescue operations requests.

All goal-oriented models share the same problem concerning the intentions operationalization. The MAP model highlights this problem by proposing several kind of guidelines to guide the user through the navigation in the map and to explain how a specific intention can be realised with a specific strategy (IAG: intention achievement guideline). This guideline can be described in several ways: in natural language, with an algorithm, through a workflow, etc. However, it is always difficult to offer an automatic way of operationalizing this guideline. Statecharts open an essential dimension to this problem by offering an automatic operationalization of the process contained in the IAG.

4 Statecharts semantics for MAP

In this section we propose a procedure for transforming a MAP to a Statechart model by using the example of the flood management process, and discuss the advantages that can be gained by this transformation.

As introduced in Section 3, a MAP is specified as a set of *intentions* to be achieved, and *strategies* for achieving them. The *intentions* can be interpreted as sets of states a process manager desires to reach. However, reaching a state does not necessarily mean that the intention is achieved. Some goals may require a number of actions to be performed before being achieved. Furthermore, taking an action aimed at attaining the goal, does not necessarily end in achieving this goal, but should reach a state which is closer to that goal than the previous one. As a result, *intentions* in statechart representation have a ”core”, which is a set of states where some actions towards attaining the goal are performed. This is a statechart representation of *strategies*.

The MAP flood management model transformed in statecharts semantics is shown in Fig. 3.

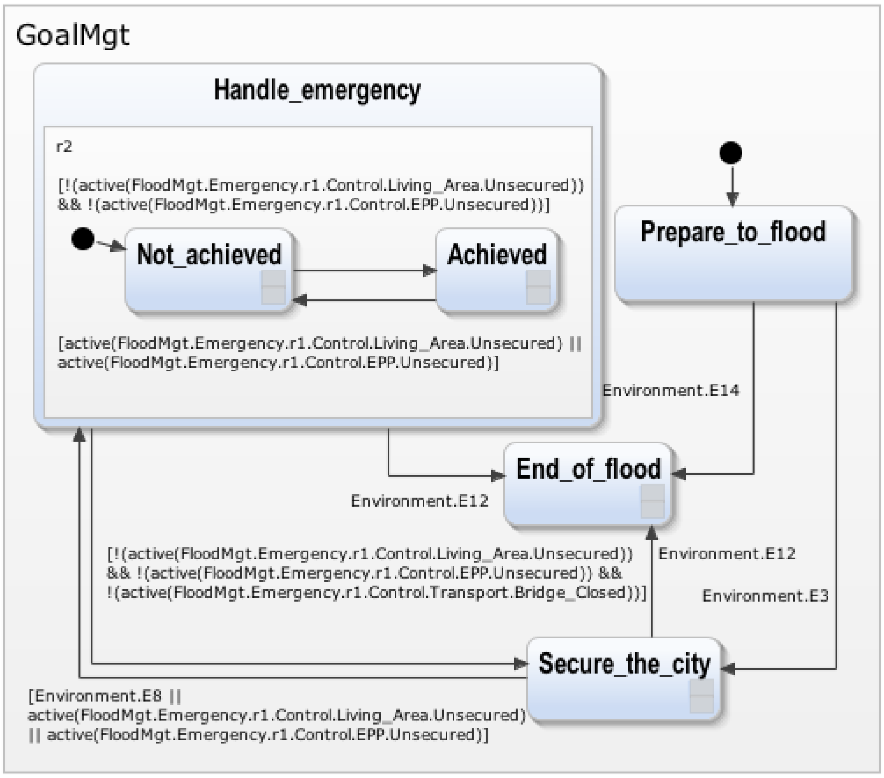

Fig. 3. Statechart representation of a MAP flood management model

For example, the condition that triggers the *Handle emergency* state is either water rises higher than 45 cm (which implies closing roads/bridges), or the *Living Area* or *EPP* sub-state becomes *Unsecured* (Fig. 1). Following the statechart model, the *Living Area* goes to *Unsecured* state when lack of resources is detected. The *EPP* sub-state, in its turn, can only reach the *Unsecured* state when the EPP is flooded and needs to be secured. Hence, lack of resources, need in closing roads/bridges or a flooded EPP force process manager to change his/her intention from *Secure the city* towards *Handle emergency*. In order to fulfill the goal and leave the state, the process manager has to request reinforcements, close roads/bridges or secure EPP. In other words, he/she has to make a decision, which *strategy* to use.

In our example of crisis, the decision-making process in statecharts can easily be operationalized.

Each state of the statechart is associated with the list of mandatory and optional activities that must/can be carried out upon entering, upon exiting and while in this state. With the state-oriented paradigm, the objective of the flood management process is as follows: the process participants (i.e., MES and Police Taskforce) should respond to the events that occur in the environment (e.g., rise of water, flooded EPP, etc.) by executing the operation procedures and producing the outcomes in order to maintain the secure functioning of the city in specified domains.

Thus, transforming a MAP representation to a Statechart representation enables operationalization of MAP and, therefore, linking them to the process scenarios for further simulation.

5 Process simulation with Yakindu Statecharts Tools

In this section, we consider the executable level of scenarios of the flood management process by simulating statechart model in YAKINDU Statecharts tool.

Fig. 4 illustrates the simulation process of the Statechart model (Fig. 1) and Statechart representation of a MAP model (Fig. 3). For clarity, the bottom right model represents a water level detector.

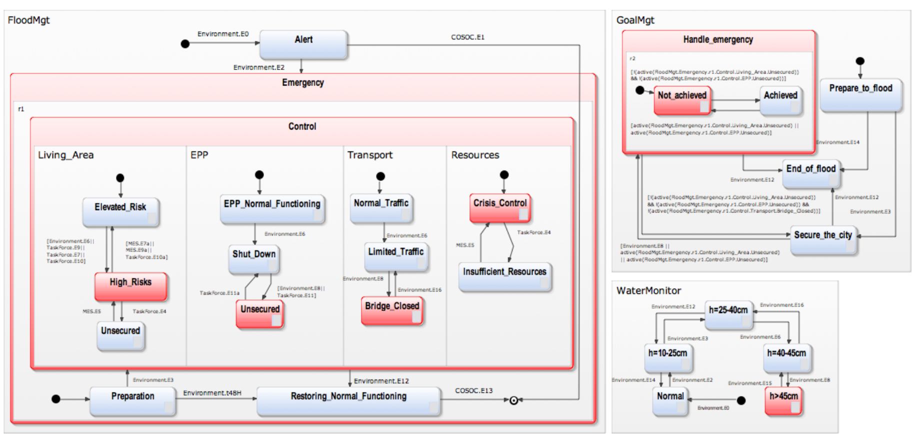

Fig. 4. Simulation of the flood management process in YAKINDU SCT

The statecharts are executed concurrently. Current situation is described by the six active (red) states: the water level is above 45 cm; the active intention is *Handle emergency*, which is not yet achieved; extremely high water level triggers a set of procedures in *High risk* state in *Living Area*; the EPP is flooded and *Unsecured* at the moment; roads/bridges are closed; resources are under *Crisis Control*.

In order to fulfill the *Handle emergency* intention and return to *Secure the city*, a process manager needs to provide EPP with security, so that the *Unsecured* state would not be active.

The state-oriented paradigm allows for *deferred activity planning*: an activity can be defined at run time, based on the desired outcome and on the context (i.e., resources, etc.). In response to unforeseen conditions, the process manager can select the next best step from the list of available activities. Thanks to *deferred binding*, he/she can also define a new activity better adapted for a situation.

Thus, the state-oriented paradigm creates a recommendation system where the process manager plays the leading role in scenario definition.

6 Conclusion

Crisis management process is safety-critical its failure could result in loss of life, significant property or environment damage. To ensure safety and security, the activities performed during crisis management are highly regulated at federal level. However, crisis handling requires high agility and never follows the same scenario.

Our experience with COSOC processes shows that a concrete flood management process relies a lot on the experience and decisions of the process manager. Assessment of a situation, adaptive scenario planning and handling the unpredictable situations represent challenges for the supporting information system.

By combining the MAP and Statecharts we arrived at a formalism which defines how human intentions drive a process. The result is an intention-driven approach to process modeling that offers an intention operationalization and provides a process manager with guidelines from goals identification to scenarios execution.

In our future work, we intend to test this process modeling approach on other cases and implement it as a part of recommendation system within COSOC.

References

1. D. Amyot and G. Mussbacher. Urn: towards a new standard for the visual description of requirements. In *Proceedings of 3rd Int. WS on Telecommunications and beyond: the broader applicability of SDL and MSC*, 2002.
2. J. Barrios and S. Nurcan. Model driven architectures for enterprise information systems. In *The 16th International Conference CAiSE'04*. Springer Verlag Lettonie, 2004.
3. A. Dardenne, A. van Lamsweerde, and S. Fickasu. Goal-directed requirements acquisition. In *Sci. Comput. Program*, volume 20. 1993.
4. R. Deneckre, I. Rychkova, and S. Nurcan. Modeling the role variability in the map process model. In *Proceedings of Fifth International Conference on Research Challenges in Information Science (RCIS)*, pages 1–9. IEEE, 2011.

5. C. Di Ciccio, A. Marrella, and A. Russo. Knowledge-intensive processes: Characteristics, requirements and analysis of contemporary approaches. *Data Semantics*, pages 1–29, 2014.
6. J. Dietz. Basic notions regarding business processes and supporting information systems. In *Proceedings of BPMDS'04, CAiSE'04 workshops proceedings*, volume 2, pages 160–168. Springer, 2004.
7. D. Harel. Statecharts: A visual formalism for complex systems. In *Science of computer programming*, volume 8, pages 231–274. 1987.
8. A. Helvas, M. Badanin, R. Pashkov, E. Kushnareva, and O. Medovik. Cos operation center. http://cosoc.ru/home, 2012.
9. E. Kornyshova, R. Deneckere, and C. Salinesi. Method chunks selection by multicriteria techniques: an extension of the assembly-based approach. In *ME 07, Switzerland*, 2007.
10. E. Kushnareva, I. Rychkova, and B. Le Grand. Modeling business processes for automated crisis management support: lessons learned. In *IEEE 9th International Conference on Research Challenges in Information Science (RCIS)*, 2015.
11. C. Rolland. A comprehensive view of process engineering. In *The 10th International Conference CAiSE'98*. Springer, 1998.
12. C. Rolland. Capturing system intentionality with maps. In *Conceptual modeling in Information systems engineering*. 2007.
13. C. Rolland and N. Prakash. Systems design for requirements expressed as a map. In *The conference IRMA'06*, 2006.
14. C. Rolland, N. Prakash, and A. Benjamen. A multi-model view of process modelling. In *Requirements engineering*, volume 4, pages 169–187. Springer-Verlag London Ltd., 1999.
15. e. The ministry of the Russian Federation for civil defense and elimination of consequences of natural disasters. Emergency management guidelines. http://www.mchs.gov.ru/, 2013.
16. A. van Lamsweerde. Reasoning about alternative requirements options. *Conceptual Modeling: Foundations and Applications*, pages 380–397, 2009.
17. E. Yu. *Modelling Strategic Relationships for Process Reengineering*. University of Toronto, Dpt of Computer Science, 1995.
18. E. Yu and J. Mylopoulos. Using goals, rules, and methods to support reasoning in business process reengineering. *International Journal of Intelligent Systems in Accounting, Finance and Management*, pages 1–13, 1996.